CTX-Configuration-Store Deployment Plan

Contents

Versions

Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
28/05/2019	1.0	First release
14/06/2019	1.1	Updated documentation for Replication steps.
02/10/2019	1.2	Updated Release

Module Versions

This version of the CTX-Configuration-Store-Install deployment plan is relevant up to version 1.1 of the CTX-Configuration-Store-Install module.

Preface

About this Manual

This document provides a guide on how to deploy the CTX-Config-Management module in your Cortex system.

Audience

This document is intended for those who require the use of CTX-Config-Management module.

Related Material

Document
CTX-Config-Management – User Guide
CTX-Config-Management.studiopkg
CTX-Config-Management – Create DB.sql
CTX-Config-Management – Create Schema.sql

Commented [JL1]: Just one script, as well as migration?

Abbreviations used in this Document

SQL Structured Query Language

SSMS SQL Server Management Studio

1 Requirements

This document details all the steps required to deploy the CTX-Config-Management module.

Requirements:

- Administrator access to Cortex Gateway
- Minimum Cortex v6.4 installed on the Cortex Application Server
- Access to the SQL Server instance
- SQL user must have SysAdmin authorisation on the SQL Server instance
- Minimum SQL Server 2012 (version 11.0.7001.0) installed on the Cortex Database Server

2 Import CTX-Configuration-Store

To deploy the CTX-Configuration-Store module on your Cortex system, CTX-Configuration-Store Studio Package needs to be imported on your Cortex system. To do this:

- Download the CTX-Configuration-Store Studio Package
- Import the Studio Package in Cortex Gateway
- Ensure the relevant users have the required permissions in 'Studio Authorisation'

After this, all users in the authorised groups will be able to view and execute the subtasks.

3 Deploy Cortex-ConfigStore Database

3.1 Overview

For the CTX-Config-Management module to work, the Cortex-ConfigStore database along with the schema must exist on the server where the Cortex databases exist. The following steps instruct you how to deploy the database and schema.

Note that if you are migrating from the old Configuration DB (v0.1), follow the steps outlined in the **Migrating (Old Version)** Section.

3.2 Create Database

1 Open Remote Desktop Connection to the Cortex database server

2 Copy the 'CTX-Configuration-Store-Install.sql' script to the Cortex database server

3 Open 'CTX-Configuration-Store-Install.sql' in SQL Server Management Server (SSMS) and connect to the DB engine where the query should be executed (this is where Cortex DBs are hosted).

   a. For a Replicated setup this step must be done on both the primary and secondary site.

4 Replace the SQL command variables as required:

   a. The variables highlighted in green must change for each sight. An example is available in the **Configuring Replication** (optional).

```
:setvar CortexDBUser "domain\cortex_cerberusDB"
:setvar DatabaseFilePath "C:\Databases"
:setvar DatabaseLogPath "C:\Databases"
:setvar Distribution_DataPath "C:\Databases"
:setvar Distribution_LogPath "C:\Databases"
:setvar InstanceName "PrimarySite"
:setvar MachineName "PrimarySite"
:setvar REPL_Admin_User "domain\cortex_sqladmin"
:setvar REPL_Working_Directory "C:\Databases"
:setvar ResilientInstanceName "SecondarySite"
:setvar DatabaseName "Cortex-ConfigStore"
:setvar DefaultFilePrefix "Cortex-ConfigStore"
:setvar DefaultDataPath ""
:setvar DefaultLogPath ""
```

Variable	Description
CortexDBUser	The Database Interface User for Cortex, e.g. domain\Cortex_CerberusDB
DatabaseFilePath	The Database Datafile Path. This must contain the following sub-folders: • <Database Name> ○ Datafile

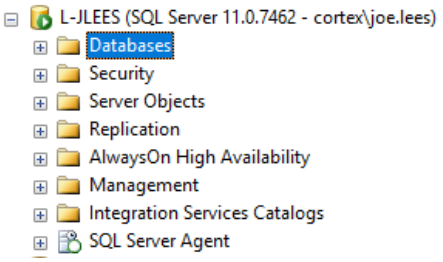
DatabaseLogPath	The Database Logfile Path. This must contain the following sub-folders: • <Database Name> ○ Logfile
Distribution_DataPath	The Distribution DB Datafile Path. This must contain the following sub-folders: • distribution ○ Datafile
Distribution_LogPath	The Distribution DB Logfile Path. This must contain the following sub-folders: • distribution ○ Logfile
InstanceName	The SQL Instance Name. See notes for Replication Setup.
MachineName	The Machine Name. See notes for Replication Setup.
REPL_Admin_User	The Admin User to handle Replication Jobs, e.g. domain\Cortex_SQLAdmin
REPL_Working_Directory	The Replication Working Directory (for Shared Drive). This must contain a folder called 'Replication Data'.
ResilientInstanceName	The Instance Name of the Resilient Server (if replication is used).
DatabaseName	The Configuration store database name. It is advised to leave the default value 'Cortex-ConfigStore'.
DefaultFilePrefix	The Configuration store database name. It is advised to leave the default value 'Cortex-ConfigStore'.
DefaultDataPath	Not used.
DefaultLogPath	Not used.

5 Create the relevant folder structure, as described in the table above. If Replication is required then the 'Replication Data' and 'distribution' folders must exist

6 Click on ***Query*** -> ***SQLCMD Mode*** and execute the query

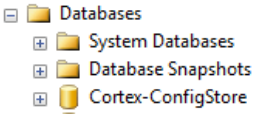
7 On the messages panel, you should see no errors on the messages and the below text

8 In the left-hand panel, click the plus to the left of 'Databases' to expand 'Databases'



9 Right click 'Databases' and click 'Refresh'.

10 Validate the 'Cortex-ConfigStore' database has been created.



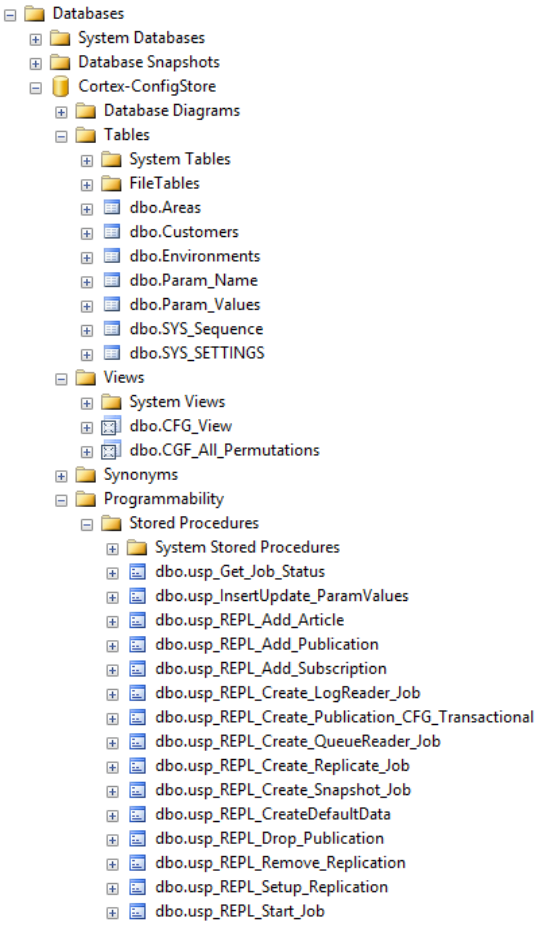
11 Expand 'Cortex-ConfigStore'

12 Expand 'Tables'

13 Expand 'Views'

14 Expand 'Programmability > Stored Procedures'

15 Validate the tables, views and stored procedures shown below are present



16 If setting up for Replication, repeat these steps on the Replicated (secondary) site

4 Configuring Replication (optional)

4.1 Example SQLCMD Variables

The following table shows an example for configuring a Replicated Setup. In this example there are 2 SQL Servers (Site-A and Site-B) each with a default SQL Instance. The domain in use is 'CTXExample' with 2 SQL Users – Cortex_CerberusDB and Cortex_SQLAdmin. On both sites the Databases should be under C:\Databases, with individual folders for distribution, Cortex-ConfigStore, and the Replicated Data folder (with all subfolders configured as mentioned in the '**Create Database**' section).

Variable	Site A Configuration	Site B Configuration
CortexDBUser	CTXExample\Cortex_CerberusDB	CTXExample\Cortex_CerberusDB
DatabaseFilePath	C:\Databases	C:\Databases
DatabaseLogPath	C:\Databases	C:\Databases
Distribution_DataPath	C:\Databases	C:\Databases
Distribution_LogPath	C:\Databases	C:\Databases
InstanceName	Site-A	Site-B
MachineName	Site-A	Site-B
REPL_Admin_User	CTXExample\SQLAdmin	CTXExample\SQLAdmin
REPL_Working_Directory	C:\Databases	C:\Databases
ResilientInstanceName	Site-B	Site-A
DatabaseName	Cortex-ConfigStore	Cortex-ConfigStore
DefaultFilePrefix	Cortex-ConfigStore	Cortex-ConfigStore

4.2 Steps to Configure (Post-Deploy)

Before configuring the Replication, the 'Replicated Data' folder must be set up as a shared drive. To do this, right click on the folder and click on the Shared tab. Ensure that it is set to read/write, as this will allow both SQL Servers to communicate with each other via the Shared Folder.

Commented [JL2]: Check all this

Once the Config Store has been deployed on both database, the following steps should be followed in order (under the context of the Config Store Database):

Site	Command
Site-A	`exec [dbo].[usp_REPL_Setup_Replication]` This sets up the replication / distribution DB and the linked servers on Site A.
Site-B	`exec [dbo].[usp_REPL_Setup_Replication]` This sets up the replication / distribution DB and the linked servers on Site B.
Site-A	`exec [dbo].[usp_REPL_Create_Publication_CFG_Transactional]` This adds the relevant data to Replication on Site A
Site-B	`exec [dbo].[usp_REPL_Create_Publication_CFG_Transactional]` This adds the relevant data to Replication on Site B

4.3 Testing Replication

First, the jobs should be checked to ensure no failures. This can be done by expanding SQL Server Agent and double clicking on Job Activity Monitor on each DB, and looking at the Replication-related jobs.

Site	Action
Site-A	Check all required objects and jobs. Insert data and check on other side.
Site-B	Check all required objects and jobs. Insert data and check on other side.

If there are any issues with the replication, there are troubleshooting steps listed in the **Troubleshooting Replication** section.

4.4 Removing Replication

Site	Command
Site-A	`exec [dbo].[usp_REPL_Drop_Publication] 'All Tables'` Removes the Publication from Site-A
Site-B	`exec [dbo].[usp_REPL_Drop_Publication] 'All Tables'` Removes the publication from Site-B
Site-A	`exec [dbo].[usp_REPL_Remove_Replication]` Removes the Replication and all related Jobs / Objects from Site-A
Site-B	`exec [dbo].[usp_REPL_Remove_Replication]` Removes the Replication and all related Jobs / Objects from Site-B
Site-A	Check all required objects and jobs are removed
Site-B	Check all required objects and jobs are removed

5 Troubleshooting Replication

5.1 Jobs

If there are issues in the replication, the first step is to check the Jobs. These can be checked under SQL Server Agent -> Job Activity Monitor by then right-clicking the relevant job and selecting 'View History'.

5.1.1 Relevant Jobs

These all assume the default name of 'Cortex-ConfigStore' has been selected.

Job Name	Details	Steps
Cortex-ConfigStore Replication - All Tables	This should always be running and handles the Replication of data	If the job is not running then it should be started manually. Any logs should be checked, and if required it can be stopped and re-started.
Cortex-ConfigStore Replication – Transaction Log Reader	This should always be running	If the job is not running it should be started.
Cortex-ConfigStore Replication – Transaction Queue Reader	This should always be running	If the job is not running it should be started.
Cortex-ConfigStore Replication – Snapshot All Tables	This is only used when a new snapshot is generated	N/A

5.2 Re-Create Replication

If there are still issues with the replication after checking the jobs, the Replication should be dropped and then re-created. To do this, follow the steps in the **Removing Replication** section followed by the steps in the **Steps to Configure (Post-Deploy)** section, and then test the Replication again.

6 Migrating (Old Version)

It is possible to migrate from the old version of the Configuration Store to the latest version. The steps are outlined below:

6.1 Pre-Requisites:

Before beginning this process, back up your existing Config Store Database.

Note that migration will replace your existing Database in the existing server location. For this reason, some of the SQLCMD Variables such as 'DatabaseFilePath' will not be relevant.

Before running any scripts, ensure that the context of the query window is set to the Configuration Database.

6.2 Migration Steps

1. Paste the CTX-Configuration-Store-Database-Migration-v01-to-v1.sql migration script into the SQL Query Window and ensure SQLCMD mode is enabled
2. Fill in variables at the top of the script. See the **Create Database** section for reference
3. Run the script to move the old data to a temporary location, create the new schema, and transfer the old data from the temporary location to the new Database.
    a. Once run, ensure that no error messages are displayed on the SQL Server Results panel
4. Validate that the data is migrated and correct, and the table objects are all in place. See **Create Database** section for details.
5. Run the script to delete old data
    a. CTX-Configuration-Store-Database-Migration-v01-Complete-Removal.sql
    b. Once run, ensure that no error messages are displayed on the SQL Server Results panel.

6.3 Rollback

In case of any issues, there is also a script provided to roll back to the old version. This should **NOT** replace a proper backup, which should always be taken manually before migrating.

See the following script: CTX-Configuration-Store-Database-Rollback-v1-to-v01.sql

